

# Finite Bisimulations for Switched Linear Systems

Ebru Aydin Gol, Xuchu Ding, Mircea Lazar and Calin Belta

**Abstract**—In this paper, we consider the problem of constructing a finite bisimulation quotient for a discrete-time switched linear system in a bounded subset of its state space. Given a set of observations over polytopic subsets of the state space and a switched linear system with stable subsystems, the proposed algorithm generates the bisimulation quotient in a finite number of steps with the aid of sublevel sets of a polyhedral Lyapunov function. Starting from a sublevel set that includes the origin in its interior, the proposed algorithm iteratively constructs the bisimulation quotient for any larger sublevel set. The bisimulation quotient can then be further used for synthesis of the switching law and system verification with respect to specifications given as syntactically co-safe Linear Temporal Logic formulas over the observed polytopic subsets.

## I. INTRODUCTION

In recent years, there has been a trend to bridge the gap between control theory and formal methods. Control theory allows for analysis and control of “complex” dynamical systems with infinite state spaces, such as systems of controlled differential equations, against “simple” specifications, such as stability and reachability. In formal methods, “simple” systems, such as finite transition systems, are checked against “complex” (rich and expressive) specification languages, such as temporal logics. Recent studies show that certain classes of dynamical systems can be abstracted to finite transition systems. Applications in robotics [1], multi-agent control systems [2], and bioinformatics [3] show that model checking and automata games can be used to analyze and control systems with non-trivial dynamics from specifications given as temporal logic formulas.

In this paper, we focus on switched linear systems made of stable subsystems, and show that a finite bisimulation abstraction of the system can be efficiently constructed within some relevant, bounded subset of the state space. Since the bisimulation quotient preserves all properties that are expressible in frameworks as rich as  $\mu$ -calculus, and implicitly Computation Tree Logic (CTL) and Linear Temporal Logic (LTL) (see *e.g.*, [4]–[6]), it can be readily used for system verification and controller synthesis against such specifications. We show how our method can be used for both controller synthesis and verification from specifications given as arbitrary formulas of a fragment of LTL, called syntactically co-safe LTL (scLTL). For controller synthesis, we

find the largest set of initial states and switching sequences such that all system trajectories satisfy a given formula. For verification, we find the largest set of initial states such that all system trajectories satisfy the formula under arbitrary switching.

The concept of constructing a finite quotient of an infinite system has been widely studied, *e.g.*, [7]–[9]. It is known that finite state bisimulation quotients exist only for specific classes of systems (*e.g.*, timed automata [9] and controllable linear systems [7]), and the well known bisimulation algorithm [4] in general does not terminate [10]. Approximately bisimilar finite abstractions for continuous-time switched systems were constructed under incremental stability assumptions in [11]. For piecewise linear systems, guided refinement procedures were employed with the goal of constructing the quotient system for verification of certain properties [8], [10].

We propose to obtain a finite bisimulation quotient of the system in a computationally feasible manner by only considering the system behavior within a relevant state space that does not contain the origin, *i.e.*, in between two positively invariant compact sets that contain the origin. Our approach relies upon the existence of a *polyhedral* Lyapunov function, which is a necessary condition for stability under arbitrary switching, see, *e.g.*, [12]. We propose to partition the state space by using sublevel sets of the Lyapunov function. Such sublevel sets, which are polytopic, allow us to generate the bisimulation quotient incrementally as the abstraction algorithm iterates, with no “holes” in the covered state space. Since we can obtain polytopic sublevel sets of any size from the Lyapunov function, the balance between the size of the abstracted state space and the amount of computation can be easily adjusted and controlled. Starting from the observation that the existence of the Lyapunov function renders the origin asymptotically stable for the switched system, its trajectories can only spend a finite time in the region of interest. As a result, we restrict our attention to LTL specifications that can be satisfied in finite time, such as scLTL formulas.

This paper is a natural, but non-trivial extension of our recent work [13], in which we used polytopic sublevel sets to generate a bisimulation quotient for a discrete autonomous linear system. Another conceptually related work is [14], where  $n$  Lyapunov functions were used for the abstraction of  $n$ -dimensional continuous-time Morse-Smale systems (*e.g.*, hyperbolic linear systems) to timed automata. The abstraction proposed therein is weaker than bisimulation, but it can be used to verify safety properties. While both [14] and this work use sublevel sets for abstraction, the main difference between [14] and this approach comes from

This work was partially supported by the NSF under grants CNS-0834260 and CNS-1035588 and by the ONR under grant MURI N00014-09-1051 at Boston University, and by Veni grant 10230 at Eindhoven University of Technology.

Ebru Aydin Gol (ebru@bu.edu) and Calin Belta (cbelta@bu.edu) are with Boston University. Xuchu Ding (dingx@utrc.utd.edu) is with United Technologies Research Center. Mircea Lazar (m.lazar@tue.nl) is with Eindhoven University of Technology.

the usage of *polyhedral Lyapunov functions*, and therefore different classes of systems for which the methods apply. Our approach removes the need for multiple orthogonal Lyapunov functions, and we argue that it allows for a more tractable implementation since the abstraction of timed automata is expensive by itself [9], and polytopic sublevel sets ensure that the abstraction algorithm requires only polytopic operations.

The rest of the paper is organized as follows. We introduce preliminaries in Sec. II and formulate the problem in Sec. III. We present the algorithm to generate the bisimulation quotient in Sec. IV, and we show in Sec. V how the resulting bisimulation quotient can be used to synthesize switching control laws and verify the system behavior against temporal logic formulas. Conclusions are summarized in Sec. VI.

## II. PRELIMINARIES

For a set  $S$ ,  $\text{int}(S)$ ,  $|S|$ , and  $2^S$  stand for its interior, cardinality, and power set, respectively. For  $\lambda \in \mathbb{R}$  and  $S \subseteq \mathbb{R}^n$ , let  $\lambda S := \{\lambda x \mid x \in S\}$ . We use  $\mathbb{R}$ ,  $\mathbb{R}_+$ ,  $\mathbb{Z}$ , and  $\mathbb{Z}_+$  to denote the sets of real numbers, non-negative reals, integer numbers, and non-negative integers. For  $m, n \in \mathbb{Z}_+$ , we use  $\mathbb{R}^n$  and  $\mathbb{R}^{m \times n}$  to denote the set of column vectors and matrices with  $n$  and  $m \times n$  real entries. For a vector  $v$  or a matrix  $A$ , we denote  $v^\top$  or  $A^\top$  as its transpose, respectively.

For a vector  $x \in \mathbb{R}^n$ ,  $[x]_i$  denotes the  $i$ -th element of  $x$  and  $\|x\|_\infty = \max_{i=1, \dots, n} |[x]_i|$  denotes the infinity norm of  $x$ , where  $|\cdot|$  denotes the absolute value. For a matrix  $Z \in \mathbb{R}^{l \times n}$ , let  $\|Z\|_\infty := \sup_{x \neq 0} \frac{\|Zx\|_\infty}{\|x\|_\infty}$  denote its induced matrix infinity norm.

A  $n$ -dimensional *polytope*  $\mathcal{P}$  (see, e.g., [15]) in  $\mathbb{R}^n$  can be described as the convex hull of  $n+1$  affinely independent points in  $\mathbb{R}^n$ . Alternatively,  $\mathcal{P}$  can be described as the intersection of  $k$ , where  $k \geq n+1$ , closed half spaces, i.e., there exists  $k \geq n+1$  and  $H_{\mathcal{P}} \in \mathbb{R}^{k \times n}$ ,  $h_{\mathcal{P}} \in \mathbb{R}^k$ , such that

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid H_{\mathcal{P}}x \leq h_{\mathcal{P}}\}. \quad (1)$$

We assume polytopes in  $\mathbb{R}^n$  are  $n$ -dimensional unless noted otherwise. The set of boundaries of a polytope  $\mathcal{P}$  are called *facets*, denoted by  $f(\mathcal{P})$ , which are themselves  $(n-1)$ -dimensional polytopes. A *semi-linear* set (also called a *polyhedron* in literature) in  $\mathbb{R}^n$  is defined as finite unions, intersections and complements of sets  $\{x \in \mathbb{R}^n \mid a^\top x \sim b, \sim \in \{=, <\}\}$ , for some  $a \in \mathbb{R}^n$  and  $b \in \mathbb{R}$ . Note that a convex and bounded semi-linear set is equivalent to a polytope with some of its facets removed.

### A. Transition systems and bisimulations

**Definition 2.1:** A transition system (TS) is a tuple  $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$ , where

- $Q$  is a (possibly infinite) set of states;
- $\Sigma$  is a set of inputs;
- $\rightarrow \subseteq Q \times \Sigma \times Q$  is a set of transitions;
- $\Pi$  is a finite set of observations; and
- $h : Q \rightarrow 2^\Pi$  is an observation map.

We denote  $x \xrightarrow{\sigma} x'$  if  $(x, \sigma, x') \in \rightarrow$ . We assume  $\mathcal{T}$  to be non-blocking, i.e., for each  $x \in Q$ , there exists  $x' \in Q$  and  $\sigma \in \Sigma$  such that  $x \xrightarrow{\sigma} x'$ . An *input word* is defined as an

infinite sequence  $\sigma = \sigma_0 \sigma_1 \dots$  where  $\sigma_k \in \Sigma$  for all  $k \in \mathbb{Z}_+$ . A *trajectory* of  $\mathcal{T}$  produced by an input word  $\sigma = \sigma_0 \sigma_1 \dots$  and originating at state  $x_0$  is an infinite sequence  $\mathbf{x} = x_0 x_1 \dots$  where  $x_k \xrightarrow{\sigma_k} x_{k+1}$  for all  $k \in \mathbb{Z}_+$ . A trajectory  $\mathbf{x}$  generates a word  $\mathbf{o} = o_0 o_1 \dots$ , where  $o_k = h(x_k)$  for all  $k \in \mathbb{Z}_+$ .

The TS  $\mathcal{T}$  is *finite* if  $|Q| < \infty$  and  $|\Sigma| < \infty$ , otherwise  $\mathcal{T}$  is *infinite*. Moreover,  $\mathcal{T}$  is *deterministic* if  $x \xrightarrow{\sigma} x'$  implies that there does not exist  $x'' \neq x'$  such that  $x \xrightarrow{\sigma} x''$ ; otherwise,  $\mathcal{T}$  is called *non-deterministic*. Given a set  $X \subseteq Q$ , we define the set of states  $\text{Pre}_{\mathcal{T}}(X, \sigma)$  that reach  $X$  in one step when input  $\sigma$  is applied as

$$\text{Pre}_{\mathcal{T}}(X, \sigma) := \{x \in Q \mid \exists x' \in X, x \xrightarrow{\sigma} x'\}. \quad (2)$$

States of a TS can be related by a relation  $\sim \subseteq Q \times Q$ . For convenience of notation, we denote  $x \sim x'$  if  $(x, x') \in \sim$ . The subset  $X \subseteq Q$  is called an *equivalence class* if  $x, x' \in X \Leftrightarrow x \sim x'$ . We denote by  $Q/\sim$  the set labeling all equivalence classes and define a map  $\text{eq} : Q/\sim \mapsto 2^Q$  such that  $\text{eq}(X)$  is the set of states in the equivalence class  $X \in Q/\sim$ .

**Definition 2.2:** We say that a relation  $\sim$  is *observation preserving* if for any  $x, x' \in Q$ ,  $x \sim x'$  implies that  $h(x) = h(x')$ .

**Definition 2.3:** A *finite partition*  $P$  of a set  $S$  is a finite collection of sets  $P := \{P_i\}_{i \in I}$ , such that  $\cup_{i \in I} P_i = S$  and  $P_i \cap P_j = \emptyset$  if  $i \neq j$ . A *finite refinement* of  $P$  is a finite partition  $P'$  of  $S$  such that for each  $P_i \in P'$ , there exists  $P_j \in P$  such that  $P_i \subseteq P_j$ .

A partition naturally induces a relation, and an observation preserving relation induces a quotient TS. One can immediately verify that a refinement of an observation preserving partition is also observation preserving.

**Definition 2.4:** An observation preserving relation  $\sim$  of a TS  $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$  induces a *quotient transition system*  $\mathcal{T}/\sim = (Q/\sim, \Sigma, \rightarrow_\sim, \Pi, h_\sim)$ , where  $Q/\sim$  is the set labeling all equivalence classes. The transitions of  $\mathcal{T}/\sim$  are defined as  $X \xrightarrow{\sigma}_\sim Y$  if and only if there exists  $x \in \text{eq}(X)$  and  $x' \in \text{eq}(Y)$  such that  $x \xrightarrow{\sigma} x'$ . The observation map is defined as  $h_\sim(X) := h(x)$ , where  $x \in \text{eq}(X)$ .

**Definition 2.5:** Given a TS  $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$ , a relation  $\sim$  is a *bisimulation relation* of  $\mathcal{T}$  if (1)  $\sim$  is observation preserving; and (2) for any  $x_1, x_2 \in Q, \sigma \in \Sigma$ , if  $x_1 \sim x_2$  and  $x_1 \xrightarrow{\sigma} x'_1$ , then there exists  $x'_2 \in Q$  such that  $x_2 \xrightarrow{\sigma} x'_2$  and  $x'_1 \sim x'_2$ .

If  $\sim$  is a bisimulation, then the quotient transition system  $\mathcal{T}/\sim$  is called a *bisimulation quotient* of  $\mathcal{T}$ . In this case,  $\mathcal{T}$  and  $\mathcal{T}/\sim$  are said to be *bisimilar*. Bisimulation is a very strong equivalence relation between systems. In fact, it preserves properties expressed in temporal logics such as LTL, CTL and  $\mu$ -calculus [4]–[6]. As such, it is used as an important tool to reduce the complexity of system verification or controller synthesis, since the bisimulation quotient (which may be finite) can be verified or used for controller synthesis instead of the original system.

### B. Polyhedral Lyapunov functions

Consider an autonomous discrete-time system,

$$x_{k+1} = \Phi(x_k), \quad k \in \mathbb{Z}_+, \quad (3)$$

where  $x_k \in \mathbb{R}^n$  is the state at the discrete-time instant  $k$  and  $\Phi : \mathbb{R}^n \mapsto \mathbb{R}^n$  is an arbitrary map with  $\Phi(0) = 0$ . Given a state  $x \in \mathbb{R}^n$ ,  $x' := \Phi(x)$  is called a *successor* state of  $x$ .

**Definition 2.6:** Let  $\lambda \in [0, 1]$ . We call a set  $\mathcal{P} \subseteq \mathbb{R}^n$   $\lambda$ -*contractive* (shortly, *contractive*) if for all  $x \in \mathcal{P}$  it holds that  $\Phi(x) \in \lambda\mathcal{P}$ . For  $\lambda = 1$ , we call  $\mathcal{P}$  a *positively invariant* set.

**Theorem 2.1:** Let  $\mathcal{X}$  be a positively invariant set for (3) with  $0 \in \text{int}(\mathcal{X})$ . Furthermore, let  $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ ,  $\rho \in (0, 1)$  and  $V : \mathbb{R}^n \mapsto \mathbb{R}_+$  such that:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \forall x \in \mathcal{X}, \quad (4)$$

$$V(\Phi(x)) \leq \rho V(x), \forall x \in \mathcal{X}. \quad (5)$$

Then system (3) is asymptotically stable in  $\mathcal{X}$ .

The proof of Thm. 2.1 can be found in [16], [17].

**Definition 2.7:** A function  $V : \mathbb{R}^n \mapsto \mathbb{R}_+$  is called a *Lyapunov function* (LF) in  $\mathcal{X}$  if it satisfies (4) and (5). If  $\mathcal{X} = \mathbb{R}^n$ , then  $V$  is called a *global Lyapunov function*.

The parameter  $\rho$  is called the *contraction rate* of  $V$ . For any  $\Gamma > 0$ ,  $\mathcal{P}_\Gamma := \{x \in \mathbb{R}^n \mid V(x) \leq \Gamma\}$  is called a *sublevel set* of  $V$ .

For the remainder of this paper we consider LFs defined using the infinity norm, i.e.,

$$V(x) = \|Lx\|_\infty, \quad L \in \mathbb{R}^{l \times n}, l \geq n, l \in \mathbb{Z}_+, \quad (6)$$

where  $L$  has full-column rank. Notice that infinity norm Lyapunov functions are a particular type of polyhedral Lyapunov functions. We opted for this type of function to simplify the exposition but in fact, the proposed abstraction method applies to general polyhedral Lyapunov functions defined by Minkowski (gauge) functions of polytopes in  $\mathbb{R}^n$  with the origin in their interior.

**Proposition 2.1:** Suppose that  $L \in \mathbb{R}^{l \times n}$  has full-column rank and  $V$  as defined in (6) is a global LF for system (3) with contraction rate  $\rho \in (0, 1)$ . Then for all  $\Gamma > 0$  it holds that  $\mathcal{P}_\Gamma$  is a polytope and  $0 \in \text{int}(\mathcal{P}_\Gamma)$ . Moreover, if  $\Phi(x)$  takes values arbitrarily from a set  $\{Ax \mid A \in \mathcal{A}\}$  for some polyhedral set  $\mathcal{A} \subseteq \mathbb{R}^{n \times n}$ , then for all  $\Gamma > 0$  it holds that  $\mathcal{P}_\Gamma$  is a  $\rho$ -contractive polytope for (3).

The proof of the above result is a straightforward application of results in [12], [18].

### III. PROBLEM FORMULATION

In this paper, we consider discrete-time switched linear systems, i.e.,

$$x_{k+1} = A_{\sigma(k)}x_k, \quad \sigma(k) \in \Sigma, k \in \mathbb{Z}_+, \quad (7)$$

where  $\sigma : \mathbb{Z}_+ \rightarrow \Sigma$  is a switching sequence that selects the active subsystem from a finite index set  $\Sigma$  and  $A_i \in \mathbb{R}^{n \times n}$  is a strictly stable (i.e., Schur) matrix for all  $i \in \Sigma$ . We assume that a global polyhedral Lyapunov function (LF) of the form (6) with contraction rate  $\rho \in (0, 1)$  is known for system (7).

Let  $\mathcal{X}$  be a polytope  $\mathcal{X} := \{x \mid \|Lx\|_\infty \leq \Gamma_{\mathcal{X}}\}$  and  $\mathcal{D}$  be a polytope  $\mathcal{D} := \{x \mid \|Lx\|_\infty \leq \Gamma_{\mathcal{D}}\}$ , where  $L$  corresponds to the polytopic LF (6) of system (7) and we assume that  $0 < \Gamma_{\mathcal{D}} < \Gamma_{\mathcal{X}}$ . Note that  $\mathcal{D} \subset \mathcal{X}$  and  $0 \in \text{int}(\mathcal{D}) \subset \text{int}(\mathcal{X})$ . We call  $\mathcal{X}$  the *working set* and  $\mathcal{D}$  the *target set*. We are

interested in synthesis of control strategies and verification of the system behavior within  $\mathcal{X}$  with respect to polytopic regions in the state space, until the target set  $\mathcal{D}$  is reached (since  $\mathcal{D}$  is positively invariant, the system trajectory will be confined within  $\mathcal{D}$  after  $\mathcal{D}$  is reached).

We assume that there exists a set  $\mathcal{R}$  of polytopes indexed by a finite set  $R$ , i.e.,  $\mathcal{R} := \{\mathcal{R}_i\}_{i \in R}$ , where  $\mathcal{R}_i \subseteq \mathcal{X} \setminus \mathcal{D}$  for all  $i \in R$ , and  $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$  for any  $i \neq j$ . The set  $\mathcal{R}$  represents regions of interest in the relevant state space, and the polytopes in  $\mathcal{R}$  are considered as observations of (7). Therefore, informally, a trajectory of (7)  $x_0x_1 \dots$  produces an infinite sequence of observations  $o_0o_1 \dots$ , such that  $o_i$  is the index of the polytope in  $\mathcal{R}$  visited by state  $x_i$ , or  $o_i = \emptyset$  if  $x_i$  is in none of the polytopes.

**Example 3.1:** Consider a system as in (7),  $\Sigma = \{1, 2\}$ ,  $A_1 = \begin{pmatrix} -0.65 & 0.32 \\ -0.42 & -0.92 \end{pmatrix}$  and  $A_2 = \begin{pmatrix} 0.65 & 0.32 \\ -0.42 & -0.92 \end{pmatrix}$ . The algorithm proposed in [12] is employed to construct a global polytopic LF of the form (6), where

$$L = \begin{pmatrix} -0.0625 & 0.6815 & 0.9947 & 0.9947 \\ 1 & 1 & 0.6868 & -0.0678 \end{pmatrix}^\top,$$

and  $\rho = 0.94$ . We chose  $\Gamma_{\mathcal{X}} = 10$  and  $\Gamma_{\mathcal{D}} = 5.063$ . (see Fig. 1 for polytopes  $\mathcal{X}$ ,  $\mathcal{D}$ , and a set of polytopes  $\mathcal{R}$ .)

The semantics of the system can be formalized through an embedding of (7) into a transition system, as follows.

**Definition 3.1:** Let  $\mathcal{X}$ ,  $\mathcal{D}$ , and  $\mathcal{R} = \{\mathcal{R}_i\}_{i \in R}$  be given. The embedding transition system for (7) is a transition system  $\mathcal{T}_e = (Q_e, \Sigma, \rightarrow_e, \Pi, h_e)$  where

- $Q_e = \{x \in \mathbb{R}^n \mid x \in \mathcal{X}\}$ ;
- $\Sigma$  is the same as the index set given in Eqn (7);
- 1) If  $x \in \mathcal{X} \setminus \mathcal{D}$ , then  $x \xrightarrow{\sigma_e} x'$  if and only if  $x' = A_\sigma x$ , i.e.,  $x'$  is the state at the next time-step after applying the dynamics of (7) at  $x$  when subsystem  $\sigma$  is active;
- 2) If  $x \in \mathcal{D}$ ,  $x \xrightarrow{\sigma_e} x$  for all  $\sigma \in \Sigma$  (since the target set  $\mathcal{D}$  is already reached, we consider the behavior of the system thereafter no longer relevant);
- $\Pi = R \cup \{\Pi_{\mathcal{D}}\}$ , i.e., the set of observations is the set of labels of regions, plus the label  $\Pi_{\mathcal{D}}$  for  $\mathcal{D}$ ;
- 1)  $h_e(x) := i$  if and only if  $x \in \mathcal{R}_i$ ;
- 2)  $h_e(x) := \emptyset$  if and only if  $x \in \mathcal{X} \setminus (\mathcal{D} \cup \bigcup_{i \in R} \mathcal{R}_i)$ ;
- 3)  $h_e(x) := \Pi_{\mathcal{D}}$  if and only if  $x \in \mathcal{D}$ .

Note that  $\mathcal{T}_e$  is deterministic and it has an infinite number of states. Moreover,  $\mathcal{T}_e$  exactly captures the system dynamics under (7) in the relevant state space  $\mathcal{X} \setminus \mathcal{D}$ , since a transition of  $\mathcal{T}_e$  naturally corresponds to the evolution of the discrete-time system in one time-step. Indeed, within  $\mathcal{X} \setminus \mathcal{D}$ , the trajectory of  $\mathcal{T}_e$  produced by an input word  $\sigma$  from a state  $x \in \mathcal{X} \setminus \mathcal{D}$  is exactly the same as the trajectory of system (7) from  $x$  under the switching sequence  $\sigma$ .

The state space of  $\mathcal{T}_e$  (which is the working set  $\mathcal{X}$ ) can be naturally partitioned as

$$P_{\mathcal{X}} := \left\{ \{\mathcal{R}_i\}_{i \in R}, \mathcal{X} \setminus (\mathcal{D} \cup \bigcup_{i \in R} \mathcal{R}_i), \mathcal{D} \right\}. \quad (8)$$

The relation induced from partition  $P_{\mathcal{X}}$  is observation

preserving (see Sec. II-A). We now formulate the main problem considered in this paper.

**Problem 3.1:** Let a system (7) with a polyhedral Lyapunov function of the form (6), sets  $\mathcal{X}$ ,  $\mathcal{D}$  and  $\{\mathcal{R}_i\}_{i \in R}$  be given. Compute a finite observation preserving partition  $P$  such that its induced relation  $\sim$  is a bisimulation of the embedding transition system  $\mathcal{T}_e$ , and obtain the corresponding bisimulation quotient  $\mathcal{T}_e/\sim$ .

**Remark 3.1:** The above assumptions on the sets  $\mathcal{X}$ ,  $\mathcal{D}$ , and  $\{\mathcal{R}_i\}_{i \in R}$  are made for simplicity of presentation. The problem formulation and the approach described in the rest of the paper can be easily extended to arbitrary positively invariant sets  $\mathcal{X}$  and  $\mathcal{D}$ , i.e., not obtained as the sublevel sets of (6), by considering the largest sublevel set that is included in  $\mathcal{D}$  and the smallest sublevel set that includes  $\mathcal{X}$  ( $\Gamma_{\mathcal{D}}$  and  $\Gamma_{\mathcal{X}}$  can be made arbitrarily small and arbitrary large, respectively, so as to capture any compact relevant subset of  $\mathbb{R}^n$ ). Also, the set of polytopes of interest  $\{\mathcal{R}_i\}_{i \in R}$  can be relaxed to a finite set of linear predicates in  $x$ .

#### IV. GENERATING THE BISIMULATION QUOTIENT

Starting from a polyhedral Lyapunov function  $V(x) = \|Lx\|_{\infty}$  with a contraction rate  $\rho = (0, 1)$  as described in Sec. II-B for system (7), we first generate a sequence of polytopic sublevel sets of the form  $\mathcal{P}_{\Gamma} := \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \Gamma\}$  as follows. Recall that  $\mathcal{X} = \mathcal{P}_{\Gamma_{\mathcal{X}}}$  and  $\mathcal{D} = \mathcal{P}_{\Gamma_{\mathcal{D}}}$  for some  $0 < \Gamma_{\mathcal{D}} < \Gamma_{\mathcal{X}}$ . We define a finite sequence  $\bar{\Gamma} := \Gamma_0, \dots, \Gamma_N$ , where

$$\Gamma_{i+1} = \rho^{-1}\Gamma_i, \quad i = 0, \dots, N-2, \quad (9)$$

$\Gamma_0 := \Gamma_{\mathcal{D}}$ ,  $\Gamma_N := \Gamma_{\mathcal{X}}$ , and  $N := \arg \min_N \{\rho^{-N}\Gamma_0 \mid \rho^{-N}\Gamma_0 \geq \Gamma_{\mathcal{X}}\}$ . This choice of  $N$  guarantees that  $\mathcal{P}_{\Gamma_{N-1}}$  is the largest sublevel set defined via (9) that is a subset of  $\mathcal{X}$ . Since  $\Gamma_N$  is exactly  $\Gamma_{\mathcal{X}}$ ,  $\mathcal{P}_{\Gamma_N}$  is exactly  $\mathcal{X}$ .

The sequence  $\bar{\Gamma}$  generates a sequence of sublevel sets  $\bar{\mathcal{P}}_{\Gamma} := \mathcal{P}_{\Gamma_0}, \dots, \mathcal{P}_{\Gamma_N}$ . From the definition of the sublevel sets and  $\bar{\Gamma}$ , we have that

$$\mathcal{P}_{\Gamma_0} \subset \dots \subset \mathcal{P}_{\Gamma_N}. \quad (10)$$

Next, we define a *slice* of the state space as follows:

$$\mathcal{S}_i := \mathcal{P}_{\Gamma_i} \setminus \mathcal{P}_{\Gamma_{i-1}}, \quad i = 1, \dots, N. \quad (11)$$

For convenience, we also denote  $\mathcal{S}_0 := \mathcal{P}_{\Gamma_0}$  (although  $\mathcal{S}_0$  is not a slice in between two sublevel sets). We immediately see that the sets  $\{\mathcal{S}_i\}_{i=0, \dots, N}$  form a partition of  $\mathcal{X}$ . Note that the slices are bounded semi-linear sets (see Sec. II).

**Example 4.1 (Example 3.1 continued):** Consider the system given in Example 3.1 and  $N = 11$  in Eqn. (9). The polytopic sublevel sets  $\bar{\mathcal{P}}_{\Gamma} := \mathcal{P}_{\Gamma_0}, \dots, \mathcal{P}_{\Gamma_{11}}$  are shown in Fig. 1.

**Proposition 4.1:** Assume that the set of slices  $\{\mathcal{S}_i\}_{i=0, \dots, N}$  is obtained from a sequence  $\bar{\Gamma}$  satisfying (9). Given a state  $x$  in the  $i$ -th slice, i.e.,  $x \in \mathcal{S}_i$ , where  $1 \leq i \leq N$ , its successor state ( $x' = A_{\sigma}x$ ,  $\sigma \in \Sigma$ ) satisfies  $x' \in \mathcal{S}_j$  for some  $j < i$ .

*Proof:* From Prop. 2.1, we have that  $\mathcal{P}_{\Gamma_i}$  are  $\rho$ -contractive. By the definition of a  $\rho$ -contractive set (Def. 2.6),

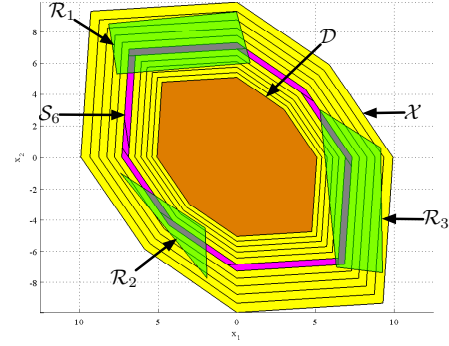


Fig. 1: An example in  $\mathbb{R}^2$  of the working set  $\mathcal{X}$ , the target set  $\mathcal{D}$  (in brown), a set of observational relevant polytopes  $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$  (in transparent green), sublevel sets with  $N = 11$  and one slice  $\mathcal{S}_0$  (in purple).

we have that  $x' = A_{\sigma}x \in \rho\mathcal{P}_{\Gamma_i} = \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \rho\Gamma_i\}$  for all  $\sigma \in \Sigma$ . From (9), we have  $\rho\Gamma_i = \Gamma_{i-1}$ . Therefore  $\mathcal{P}_{\Gamma_{i-1}} = \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \Gamma_{i-1}\}$  implies that  $\mathcal{P}_{\Gamma_{i-1}} = \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \rho\Gamma_i\}$  and hence  $\mathcal{P}_{\Gamma_{i-1}} = \rho\mathcal{P}_{\Gamma_i}$  and  $x' \in \mathcal{P}_{\Gamma_{i-1}}$ . From the definition of slices (11),  $x' \in \mathcal{S}_j$  for some  $j < i$ . ■

We now present the abstraction algorithm (see Alg. 1) that computes the bisimulation quotient. In Alg. 1, we make use of two procedures `ComputePre` and `RefineUpdate`, which will be further explained below. The main idea is to start with  $P_{\mathcal{X}}$  (Eqn. (8)), refine the partition according to  $\{\mathcal{S}_i\}_{i=0, \dots, N}$  to guarantee that it is a refinement to both  $P_{\mathcal{X}}$  as in (8) and  $\{\mathcal{S}_i\}_{i=0, \dots, N}$ , and then iteratively refine according to the `Pre` operator (see Eqn. 2). The first step, starting with  $P_{\mathcal{X}}$ , is necessary so that the partition is observation preserving. The second step guarantees that each element in the partition is included in a slice. The third step allows us to ensure that at iteration  $i$  of the algorithm, the bisimulation quotient for states within  $\mathcal{P}_{\Gamma_i}$  is completed.

The procedure `ComputePre`( $\tilde{\mathcal{P}}, \sigma$ ) takes as input  $\tilde{\mathcal{P}}$ , which is a bounded semi-linear set (e.g., a slice), and  $\sigma \in \Sigma$ , which is the switching input, and returns the set  $\text{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$ . If  $\tilde{\mathcal{P}}$  is a polytope, then  $\text{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$  is computed as

$$\text{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma) = \{x \in \mathbb{R}^n \mid H_{\tilde{\mathcal{P}}}A_{\sigma}x \leq h_{\tilde{\mathcal{P}}}\}. \quad (12)$$

In general, if  $\tilde{\mathcal{P}}$  is a semi-linear set, then  $\text{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$  is also a semi-linear set and it can be computed via quantifier elimination [19]. In particular,  $\text{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$  for a bounded semi-linear set  $\tilde{\mathcal{P}}$  can be computed via a convex decomposition and repeated applications of (12). This computation is discussed in more detail in [13]. Note that `ComputePre`( $\tilde{\mathcal{P}}, \sigma$ ) only requires polytopic operations.

The procedure `RefineUpdate`( $P, \mathcal{T}, \tilde{\mathcal{P}}, \sigma, q$ ) (outlined in Alg. 2) refines a partition  $P$  with respect to set  $\tilde{\mathcal{P}}$ , where  $\tilde{\mathcal{P}} = \text{ComputePre}(\text{eq}(q), \sigma)$ . It then updates  $\mathcal{T}$ . If  $P$  consists of only bounded semi-linear sets and  $\tilde{\mathcal{P}}$  is a semi-linear set, then the resulting refinement  $P^+$  consists of only bounded semi-linear sets. This fact allows us to always use `ComputePre`( $\tilde{\mathcal{P}}, \sigma$ ).

---

**Algorithm 1** Abstraction algorithm

---

**Input:** System dynamics (7), polytopic LF  $V(x) = \|Lx\|_\infty$  with a contractive rate  $\rho$ , sets  $\mathcal{X}$ ,  $\mathcal{D}$  and  $\{\mathcal{R}_i\}_{i \in R}$ .  
**Output:**  $\mathcal{T}_e/\sim$  as a bisimulation quotient of the embedding transition system  $\mathcal{T}_e$  and the corresponding observation preserving partition  $P$ .

- 1: Obtain  $P_{\mathcal{X}}$  as in (8).
- 2: Generate the sequence of sublevel sets  $\tilde{P}_\Gamma = \mathcal{P}_{\Gamma_0}, \dots, \mathcal{P}_{\Gamma_N}$  and slices  $\mathcal{S}_0, \dots, \mathcal{S}_N$  as defined in (11).
- 3: Set  $P_0 = \{\emptyset \subset \tilde{P}_1 \cap \tilde{P}_2 \mid \tilde{P}_1 \in P_{\mathcal{X}}, \tilde{P}_2 \in \{\mathcal{S}_i\}_{i=0, \dots, N}\}$ .
- 4: Initialize  $\mathcal{T}_e/\sim_0$  by setting  $Q_e/\sim_0$  as the set labeling  $P_0$ . Set transitions only for the state  $q \in Q_e/\sim_0$  where  $\text{eq}(q) = \mathcal{S}_0 = \mathcal{D}$  with  $q \xrightarrow{\sigma} q$  for all  $\sigma \in \Sigma$ .
- 5: **for** each  $i = 0, \dots, N-1$  **do**
- 6:   Set  $\mathcal{T}_e/\sim_{i+1} = \mathcal{T}_e/\sim_i$  **and**  $P_{i+1} = P_i$ .
- 7:   **for** each  $q \in Q_e/\sim_i$  where  $\text{eq}(q) \subseteq \mathcal{S}_i$  **do**
- 8:     **for** each  $\sigma \in \Sigma$  **do**
- 9:       Find  $\tilde{P} = \text{ComputePre}(\text{eq}(q), \sigma)$ .
- 10:       Set  $[P_{i+1}, \mathcal{T}_e/\sim_{i+1}] = \text{RefineUpdate}(P_{i+1}, \mathcal{T}_e/\sim_{i+1}, \tilde{P}, \sigma, q)$ .
- 11:     **end for**
- 12:   **end for**
- 13: **end for**
- 14: Return  $\mathcal{T}_e/\sim_N$  and  $P_N$  as a solution to Prob. 3.1.

---

---

**Algorithm 2**  $[P^+, \mathcal{T}^+] = \text{RefineUpdate}(P, \mathcal{T}, \tilde{P}, \sigma, q)$ 

---

**Input:** A TS  $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$ , a partition  $P$  where  $\text{eq}(q') \in P$  for all  $q' \in Q$ , and  $\tilde{P} = \text{ComputePre}(\text{eq}(q), \sigma)$  for some  $q \in Q$ ,  $\sigma \in \Sigma$ .  
**Output:**  $P^+$  is a finite refinement of  $P$  with respect to  $\tilde{P}$ ,  $\mathcal{T}^+$  is a TS updated from  $\mathcal{T}$ .

- 1: Set  $P^+ = P$  **and**  $\mathcal{T}^+ = \mathcal{T}$ .
- 2: **for all**  $q' \in Q^+$  such that  $\text{eq}(q') \cap \tilde{P} \neq \emptyset$  **do**
- 3:   Replace  $q'$  in  $Q^+$  by  $\{q_1, q_2\}$  and set  $\text{eq}(q_1) = \text{eq}(q') \cap \tilde{P}$ ,  $\text{eq}(q_2) = \text{eq}(q') \setminus \tilde{P}$ .
- 4:   Replace  $\text{eq}(q')$  in  $P^+$  by  $\{\text{eq}(q_1), \text{eq}(q_2)\}$ .
- 5:   Replace each  $(q', \sigma', q'') \in \rightarrow^+$  by  $\{(q_i, \sigma', q'')\}_{i=1,2}$ .
- 6:   Add transition  $(q_1, \sigma, q)$  to  $\rightarrow^+$ .
- 7: **end for**

---

The correctness of Alg. 1 will be shown by an inductive argument. Given a sublevel set  $\mathcal{P}_{\Gamma_i}$  and a partition  $P_i$  as obtained in Alg. 1, we define  $\tilde{P}_i$  as

$$\tilde{P}_i := \{\tilde{P} \in P_i \mid \tilde{P} \subseteq \mathcal{P}_{\Gamma_i}\}. \quad (13)$$

From Alg. 1, we see that  $P_0$  partitions all the slices, and since  $P_i$  is a finite refinement of  $P_0$ , we can directly see that  $\tilde{P}_i$  is a partition of  $\mathcal{P}_{\Gamma_i}$ . Let us define an embedding transition system  $\mathcal{T}_e(i)$  as a subset of  $\mathcal{T}_e$  with set of states  $\{x \in Q_e \mid x \in \mathcal{P}_{\Gamma_i}\}$  and let us state the following result.

**Proposition 4.2:** At the completion of the  $i$ -th iteration (of the outer loop) of Alg. 1 (where  $P_{i+1}$  is obtained), if  $\sim_i$  induced by  $\tilde{P}_i$  as defined in (13) is a bisimulation of  $\mathcal{T}_e(i)$ , then  $\sim_{i+1}$  induced by  $\tilde{P}_{i+1}$  is a bisimulation of  $\mathcal{T}_e(i+1)$ .

*Proof:* We show that at the end of  $i$ -th iteration, each transition originating at a state  $q \in Q_e/\sim_{i+1}$  with  $\text{eq}(q) \subseteq \mathcal{P}_{\Gamma_{i+1}}$  satisfies the bisimulation requirement (Def. 2.5). By Prop. 4.1, for each  $x \in \mathcal{S}_{i+1}$  and  $\sigma \in \Sigma$ ,  $x' = A_\sigma x$  must be in a slice with a lower index and thus  $x' \in \mathcal{T}_e(i)$ . Let  $x \in \text{eq}(q) \in P_i$ . If  $x' \in \mathcal{S}_i$ , then we have  $x \in \tilde{P} = \text{ComputePre}(\text{eq}(q'), \sigma)$  (from step 9 of Alg. 1) for some  $q' \in Q_e/\sim_i$ . The **RefineUpdate** procedure replaces  $\text{eq}(q)$  with  $\text{eq}(q_1) = \text{eq}(q) \cap \tilde{P}$  and  $\text{eq}(q_2) = \text{eq}(q) \setminus \tilde{P}$ , and updates  $\mathcal{T}_e/\sim_{i+1}$ . We note from Eqn. (2) that for any  $x \in \text{eq}(q_1)$ ,  $x' = A_\sigma x \in \text{eq}(q')$ , thus for any  $x_1, x_2 \in \text{eq}(q_1)$ ,  $x_1 \sim x_2$ ,  $A_\sigma x_1 \sim A_\sigma x_2$ . Moreover, the same argument holds for any subset of  $\text{eq}(q_1)$ . Therefore, the transitions given in steps 5 and 6 of Alg. 2 satisfy the bisimulation requirement. On the other hand, if  $x' \notin \mathcal{S}_i$ , then  $x' \in \mathcal{S}_j$  for some  $j < i$  and  $x$  is already in a set  $\text{eq}(q)$ , where  $q \xrightarrow{\sigma} \text{eq}(q')$  for some  $q'$  satisfying the bisimulation requirement. Therefore, step 9 of Alg. 1 provides exactly the transitions needed for all states in  $\mathcal{S}_{i+1}$  and thus,  $\sim_{i+1}$  induced by  $\tilde{P}_{i+1}$  is a bisimulation of  $\mathcal{T}_e(i+1)$ . ■

**Theorem 4.1:** Alg. 1 returns a solution to Prob. 3.1 in finite time.

*Proof:* From Alg. 2, we have that  $P_i$  is a refinement of  $P_{\mathcal{X}}$  for any  $i = 0, \dots, N$ . Therefore,  $P_N$  and its induced relation  $\sim_N$  are observation preserving.

At step 4 of Alg. 1, we set  $q \xrightarrow{\sigma} q$ ,  $\forall \sigma \in \Sigma$  where  $\text{eq}(q) = \mathcal{D}$ . From the definition of  $\mathcal{T}_e$ , we see that since  $\mathcal{D}$  is the only state,  $\sim_0$  induced by  $\tilde{P}_0$  is a bisimulation of  $\mathcal{T}_e(0)$ . Using Prop. 4.2 and induction, at iteration  $N-1$ , we have that  $\sim_N$  induced by  $\tilde{P}_N$  is a bisimulation of  $\mathcal{T}_e(N)$ . Note that  $\tilde{P}_N$  is exactly  $P_N$ ,  $\mathcal{P}_{\Gamma_N}$  is exactly  $\mathcal{X}$  and  $\mathcal{T}_e(N)$  is exactly  $\mathcal{T}_e$ . Therefore  $\sim_N$  induced by  $P_N$  is a bisimulation of  $\mathcal{T}_e$ .

Finally, note that at each iteration  $i$ , the number of updated sets is finite as the partition  $P_i$  and the set of inputs  $\Sigma$  are finite. Therefore, the bisimulation quotient is finite and Alg. 1 completes in finite time. ■

**Example 4.2 (Example 4.1 continued):** Alg. 1 is applied on the same setting as in Example 4.1 to compute the bisimulation quotient.  $P_3$  and  $P_{11}$  are shown in Fig. 2.

## V. TEMPORAL LOGIC SYNTHESIS AND VERIFICATION

After we obtain a bisimulation quotient for system (7), we can solve verification and controller synthesis problems from temporal logic specifications such as CTL\*, CTL and LTL. The asymptotic stability assumption implies that all trajectories of (7) sink in  $\mathcal{D}$ . For this reason, we will focus on syntactically co-safe fragment of LTL, which includes all specifications of LTL where satisfactions of trajectories can be determined by a finite prefix. Since we are interested in the behavior of (7) until  $\mathcal{D}$  is reached, scLTL is sufficiently rich as the specification language.

A detailed description of the syntax and semantics of scLTL is beyond the scope of this paper and can be found in, for example, [20], [21]. Roughly, an scLTL formula is built up from a set of atomic propositions  $\Pi$ , standard Boolean operators  $\neg$  (negation),  $\vee$  (disjunction),  $\wedge$  (conjunction),  $\Rightarrow$

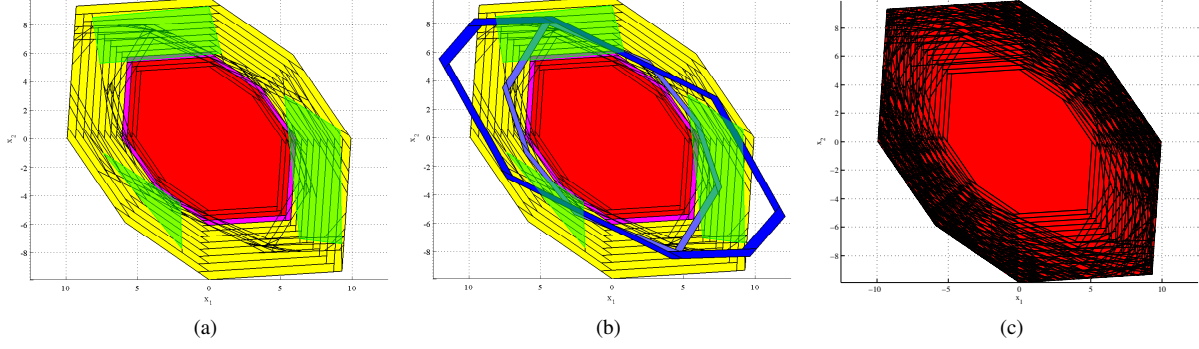


Fig. 2: The observed regions are shown in transparent green in (a) and (b). (a) At the end of the third iteration ( $i = 2$ ), the bisimulation quotient for states within  $\mathcal{P}_{\Gamma_3}$  is completed, which are shown in red and purple. In the fourth iteration, the states within  $\mathcal{P}_{\Gamma_{11}} \setminus \mathcal{P}_{\Gamma_3}$  will be partitioned according to  $\text{Pre}_{\mathcal{T}_e}(\tilde{\mathcal{P}}, \sigma)$ ,  $\tilde{\mathcal{P}} \in \mathcal{S}_3$ . (b)  $\mathcal{S}_3$  is shown in purple, and  $\text{Pre}_{\mathcal{T}_e}(\mathcal{S}_3, 1)$  and  $\text{Pre}_{\mathcal{T}_e}(\mathcal{S}_3, 2)$  are shown in light and dark blue. (c) At the last iteration where  $i = 10$ , the algorithm is completed. The state space covered by the bisimulation quotient is shown in red, covering all of  $\mathcal{X}$ .

(implication) and temporal operators  $\text{X}$  (next),  $\text{U}$  (until) and  $\text{F}$  (eventually). The semantics of scLTL formulas is given over infinite words  $\mathbf{o} = o_0 o_1 \dots$ , where  $o_i \in 2^\Pi$  for all  $i$ . We write  $\mathbf{o} \models \phi$  if the word  $\mathbf{o}$  satisfies the scLTL formula  $\phi$ . We say a trajectory  $\mathbf{q}$  of a transition system  $\mathcal{T}$  satisfies scLTL formula  $\phi$ , if the word generated by  $\mathbf{q}$  (see Def. 2.1) satisfies  $\phi$ .

**Example 5.1:** Again, consider the setting in Example 3.1 with  $\mathcal{R} = \{\mathcal{R}_i\}_{i=\{1,2,3\}}$ . We now consider a specification in scLTL over  $\{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \Pi_D\}$ . For example, the specification “A system trajectory never visits  $\mathcal{R}_2$  and eventually visits  $\mathcal{R}_1$ . Moreover, if it visits  $\mathcal{R}_3$  then it must not visit  $\mathcal{R}_1$  at the next time step” can be translated to a scLTL formula:

$$\phi := (\neg \mathcal{R}_2 \text{ U } \Pi_D) \wedge \text{F } \mathcal{R}_1 \wedge ((\mathcal{R}_3 \Rightarrow \text{X } \neg \mathcal{R}_1) \text{ U } \Pi_D) \quad (14)$$

#### A. Synthesis of switching strategies

In this section, we assume that we can choose the dynamics  $A_\sigma$ ,  $\sigma \in \Sigma$  to be applied at each step  $k$ . Our goal is to find a set of initial states and a switching sequence (i.e., a sequence of elements from  $\Sigma$  to be applied at each step) for each initial state such that all the corresponding trajectories of system (7) satisfy a temporal logic specification. Formally, we consider the following problem:

**Problem 5.1:** Consider system (7) with a polyhedral Lyapunov function in the form of (6), sets  $\mathcal{X}$ ,  $\mathcal{D}$  and  $\{\mathcal{R}_i\}_{i \in R}$ , and a scLTL formula  $\phi$  over  $R \cup \{\Pi_D\}$ . Find the largest set  $\mathcal{X}^S \subseteq \mathcal{X}$  and a function  $\Omega : \mathcal{X}^S \mapsto \Sigma^*$  such that the trajectory of system (7) initiated from a state  $x_0 \in \mathcal{X}^S$  under the switching sequence  $\Omega(x_0)$  satisfies  $\phi$ .

As a switched system is deterministic, it produces a unique trajectory for a given initial state and switching sequence. This fact allows us to provide a solution to Problem 5.1 as an assignment of a switching sequence to each initial state. Our solution to Prob. 5.1 proceeds by finding a bisimulation quotient  $\mathcal{T}_e/\sim$  of the embedding transition system  $\mathcal{T}_e$  using Alg. 1. Then we translate  $\phi$  to a Finite State Automaton (FSA), defined below.

**Definition 5.1:** A deterministic finite state automaton (FSA) is a tuple  $\mathcal{A} = (S_{\mathcal{A}}, S_{\mathcal{A}0}, \Sigma, \delta_{\mathcal{A}}, F_{\mathcal{A}})$  where

- $S_{\mathcal{A}}$  is a finite set of states;
- $S_{\mathcal{A}0} \subseteq S_{\mathcal{A}}$  is a set of initial states;
- $\Sigma$  is an input alphabet;
- $\delta_{\mathcal{A}} : S_{\mathcal{A}} \times \Sigma \rightarrow S_{\mathcal{A}}$  is a transition function;
- $F_{\mathcal{A}} \subseteq S_{\mathcal{A}}$  is a set of final states.

A word  $\sigma = \sigma_0 \dots \sigma_{d-1}$  over  $\Sigma$  generates a trajectory  $s_0 \dots s_d$ , where  $s_0 \in S_{\mathcal{A}0}$  and  $\delta(s_i, \sigma_i) = s_{i+1}$  for all  $i = 0, \dots, d-1$ .  $\mathcal{A}$  accepts word  $\sigma$  if  $s_d \in F_{\mathcal{A}}$ .

For any scLTL formula  $\phi$  over  $\Pi$ , there exists a FSA  $\mathcal{A}$  with input alphabet  $2^\Pi$  that accepts the prefixes of all and only the satisfying words [20], [22].

**Definition 5.2:** Given a transition system  $\mathcal{T} = (Q, \Sigma, \rightarrow, \Pi, h)$  and a FSA  $\mathcal{A} = (S_{\mathcal{A}}, S_{\mathcal{A}0}, 2^\Pi, \delta_{\mathcal{A}}, F_{\mathcal{A}})$ , their product automaton, denoted by  $\mathcal{PA} = \mathcal{T} \times \mathcal{A}$ , is a tuple  $\mathcal{PA} = (S_{\mathcal{PA}}, S_{\mathcal{PA}0}, \Sigma, \rightarrow_{\mathcal{PA}}, F_{\mathcal{PA}})$  where

- $S_{\mathcal{PA}} = Q \times S_{\mathcal{A}}$ ;
- $S_{\mathcal{PA}0} = Q \times S_{\mathcal{A}0}$ ;
- $\rightarrow_{\mathcal{PA}} \subseteq S_{\mathcal{PA}} \times \Sigma \times S_{\mathcal{PA}}$  is the set of transitions, defined by:  $((q, s), \sigma, (q', s')) \in \rightarrow_{\mathcal{PA}}$  iff  $q \xrightarrow{\sigma} q'$  and  $\delta_{\mathcal{A}}(s, h(q)) = s'$ ;
- $F_{\mathcal{PA}} = Q \times F_{\mathcal{A}}$ .

We denote  $s_{\mathcal{PA}} \xrightarrow{\sigma} s'_{\mathcal{PA}}$  if  $(s_{\mathcal{PA}}, \sigma, s'_{\mathcal{PA}}) \in \rightarrow_{\mathcal{PA}}$ . A trajectory  $\mathbf{p} = (q_0, s_0) \dots (q_d, s_d)$  of  $\mathcal{PA}$  produced by input word  $\sigma = \sigma_0 \dots \sigma_{d-1}$  is a finite sequence such that  $(q_0, s_0) \in S_{\mathcal{PA}0}$  and  $(q_k, s_k) \xrightarrow{\sigma_k}_{\mathcal{PA}} (q_{k+1}, s_{k+1})$  for all  $k = 0, \dots, d-1$ .  $\mathbf{p}$  is called accepting if  $(q_d, s_d) \in F_{\mathcal{PA}}$ .

By the construction of  $\mathcal{PA}$  from  $\mathcal{T}$  and  $\mathcal{A}$ ,  $\mathbf{p}$  produced by  $\sigma$  is accepting if and only if  $\mathbf{q} = \gamma_{\mathcal{T}}(\mathbf{p})$  satisfies the scLTL formula corresponding to  $\mathcal{A}$  [21], where  $\gamma_{\mathcal{T}}(\mathbf{p})$  is the projection of a trajectory  $\mathbf{p}$  of  $\mathcal{PA}$  onto  $\mathcal{T}$  by simply removing the automaton part of the state in  $s_{\mathcal{PA}} \in S_{\mathcal{PA}}$ .

We construct the product  $\mathcal{PA}$  between the quotient transition system  $\mathcal{T}_e/\sim$  obtained from Alg. 1 and FSA  $\mathcal{A}$  corresponding to specification formula  $\phi$ . By performing a graph search on  $\mathcal{PA}$ , we can find the largest subset  $S_{\mathcal{PA}}^S$  of



$S_{\mathcal{PA}}$  and a feedback control function  $\Omega_{\mathcal{PA}} : S_{\mathcal{PA}}^S \mapsto \Sigma$  such that the trajectories of  $\mathcal{PA}$  originating in  $S_{\mathcal{PA}}^S$  in closed loop with  $\Omega_{\mathcal{PA}}$  reach  $F_{\mathcal{PA}}$ . Then, we define the set of satisfying initial states of system (7) from  $S_{\mathcal{PA}}^S$  as

$$\mathcal{X}^S = \{eq(q) \mid (q, s) \in (S_{\mathcal{PA}0} \cap S_{\mathcal{PA}}^S)\}. \quad (15)$$

Since  $\mathcal{PA}$  is deterministic,  $\Omega_{\mathcal{PA}}$  defines a unique input word for each  $(q_0, s_0) \in S_{\mathcal{PA}}^S$ . Moreover, an input word of  $\mathcal{PA}$  directly maps to a switching sequence for system (7). Formally, the switching sequence  $\Omega : \mathcal{X}^S \mapsto \Sigma^*$  is obtained by “projecting”  $\Omega_{\mathcal{PA}}$  from  $\mathcal{PA}$  to  $\mathcal{T}$  as follows:

$$\Omega(x) = \Omega_{\mathcal{PA}}((q_0, s_0)) \dots \Omega_{\mathcal{PA}}((q_{d-1}, s_{d-1})), \quad (16)$$

where  $x \in eq(q_0)$ ,  $s_0 \in S_{\mathcal{A}0}$ ,  $(q_i, s_i) \xrightarrow{\Omega_{\mathcal{PA}}((q_i, s_i))} \mathcal{PA}(q_{i+1}, s_{i+1})$ , for each  $i = 0, \dots, d-1$  and  $(q_d, s_d) \in F_{\mathcal{PA}}$ .

**Proposition 5.1:**  $\mathcal{X}^S$  as defined in Eqn. (15) and function  $\Omega$  as defined in Eqn. (16) solve Prob. 5.1.

*Proof:* For each  $x \in \mathcal{X}^S$ , there exists  $(q_0, s_0) \in S_{\mathcal{PA}}^S$  such that  $x \in eq(q_0)$  and  $s_0 \in S_{\mathcal{A}0}$  by Eqn. (15). By construction of  $\mathcal{PA}$  and definition of  $\Omega$  (Eqn. (16)), the trajectory of  $\mathcal{T}_e/\sim$  originating at  $q_0$  and generated by input word  $\Omega(x)$  satisfies  $\phi$ . Then by bisimulation relation the trajectories of (7) originating in  $eq(q_0)$  and generated by switching sequence  $\Omega(x)$  satisfy  $\phi$ .

We prove that  $\mathcal{X}^S$  is the largest set of satisfying initial states by contradiction. Assume that there exists  $x_0 \notin \mathcal{X}^S$  such that a trajectory  $\mathbf{x} = x_0 \dots x_d$  originating at  $x_0$  of (7) produced by switching sequence  $\sigma = \sigma_0 \dots \sigma_{d-1}$  satisfies  $\phi$ , and  $x_0 \in eq(q_0)$  where  $q_0 \in Q_e/\sim$ . Then by the bisimulation relation 1) there exists a trajectory  $\mathbf{q} = q_0 \dots q_d$  of  $\mathcal{T}_e/\sim$  such that  $x_i \in eq(q_i)$ ,  $q_i \xrightarrow{\sigma_i} q_{i+1}$  for all  $i = 0, \dots, d-1$  and  $x_d \in eq(q_d)$ , 2)  $\mathbf{q}$  satisfies  $\phi$ . However, we know that on the product  $\mathcal{PA} = \mathcal{T}_e/\sim \times \mathcal{A}$ ,  $F_{\mathcal{PA}}$  is not reachable from  $\{(q_0, s) \mid s \in S_{\mathcal{A}0}\}$ . Hence, a trajectory  $\mathbf{p}$  originating in  $\{(q_0, s) \mid s \in S_{\mathcal{A}0}\}$  cannot be accepting on  $\mathcal{PA}$ , and by construction of  $\mathcal{PA}$  [21]  $\gamma_{\mathcal{T}_e/\sim}(\mathbf{p})$  as a trajectory of  $\mathcal{T}_e/\sim$  cannot satisfy formula  $\phi$ , which yields a contradiction. ■

**Example 5.2 (Example 5.1 continued):** For the example specification  $\phi$  (14), we obtained the solution to Prob. 5.1. The FSA has 6 states and the quotient TS obtained from Alg. 1 has 9677 states. The set of initial states  $\mathcal{X}^S$  is shown in Fig. 3.

### B. Verification under arbitrary switching

**Problem 5.2:** Consider system (7) with a polyhedral Lyapunov function in the form of (6), sets  $\mathcal{X}$ ,  $\mathcal{D}$  and  $\{\mathcal{R}_i\}_{i \in R}$ , and a scLTL formula  $\phi$  over  $R \cup \{\Pi_{\mathcal{D}}\}$ . Find the largest set  $\mathcal{X}^{AS} \subseteq \mathcal{X}$  such that all trajectories of system (7) originating in  $\mathcal{X}^{AS}$  satisfy  $\phi$  under arbitrary switching.

Note that system (7) under arbitrary switching is uncontrolled and non-deterministic, i.e., at every time-step a subsystem is arbitrarily chosen from the set  $\Sigma$ . Therefore, we define an embedding transition system  $\mathcal{T}_e^A = \{Q_e, \Sigma^A, \rightarrow_e^A, h_e\}$  for the arbitrary switching setup from the embedding transition system  $\mathcal{T}_e = \{Q_e, \Sigma, \rightarrow_e, h_e\}$  (Def. 3.1) by adapting the input set and the set of transitions as follows:

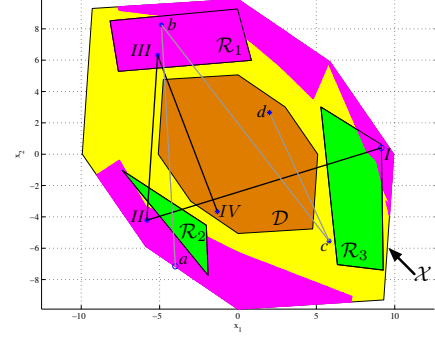


Fig. 3:  $\mathcal{X}^S$  is shown in purple.  $\mathcal{X}$ ,  $\mathcal{D}$ ,  $\{\mathcal{R}_i\}_{i \in R}$  and two sample trajectories are indicated by their labels.

- $\Sigma^A = \{\epsilon\}$ ,
- $\rightarrow_e^A = \{(q, \epsilon, q') \mid \exists \sigma \in \Sigma, (q, \sigma, q') \in \rightarrow_e\}$ .

We denote  $q \rightarrow_e^A q'$  if  $(q, \epsilon, q') \in \rightarrow_e^A$ . We use  $\epsilon$  as a “dummy” input because the transitions of  $\mathcal{T}_e^A$  are not controlled. Note that  $\mathcal{T}_e^A$  is infinite and non-deterministic. Moreover,  $\mathcal{T}_e^A$  exactly captures dynamics of system (7) under arbitrary switching in the relevant state space  $\mathcal{X} \setminus \mathcal{D}$ .

Our solution to Prob 5.2 parallels the solution we proposed for Prob. 5.1. We first convert the bisimulation quotient  $\mathcal{T}_e/\sim = \{Q_e/\sim, \Sigma, \rightarrow_e/\sim, h_e/\sim\}$  of  $\mathcal{T}_e$  obtained from Alg. 1 to  $\mathcal{T}_e^A/\sim = \{Q_e/\sim, \Sigma^A, \rightarrow_e^A/\sim, h_e/\sim\}$  as follows:

- $\Sigma^A = \{\epsilon\}$ ,
- $\rightarrow_e^A/\sim = \{(q, \epsilon, q') \mid \exists \sigma \in \Sigma, (q, \sigma, q') \in \rightarrow_e/\sim\}$ .

In this case, we have a particular bisimulation relation. The embedding and the quotient transition systems have a single input that labels all the transitions.

**Proposition 5.2:**  $\mathcal{T}_e^A/\sim$  is a bisimulation quotient of  $\mathcal{T}_e^A$ .

*Proof:* Let  $q_1, q_2 \in eq(q)$ ,  $q'_1 \in eq(q')$  and  $q_1 \rightarrow_e^A q'_1$ , where  $q, q' \in Q_e/\sim$  and  $q_1, q_2, q'_1 \in Q_e$ . To prove the bisimulation property we need to show that there exists  $q'_2 \in eq(q')$  such that  $q_2 \rightarrow_e^A q'_2$ .

If  $q_1 \rightarrow_e^A q'_1$ , then there exists  $\sigma \in \Sigma$  such that  $q_1 \xrightarrow{\sigma} q'_1$ , i.e.,  $q'_1 = A_\sigma q_1$ . Steps 9 and 10 of Alg. 1 guarantee that  $eq(q) \subseteq \text{Pre}_{\mathcal{T}_e}(eq(q'), \sigma)$ . Therefore, for all  $q_i \in eq(q)$ ,  $A_\sigma q_i \in eq(q')$ , and hence for all  $q_i \in eq(q)$ ,  $q_i \rightarrow_e^A q_j$  for some  $q_j \in eq(q')$ . ■

Parallel to our solution to Prob. 5.1, we construct a FSA  $\mathcal{A}$  corresponding to specification formula  $\phi$ , and then we take the product  $\mathcal{PA}^A = (S_{\mathcal{PA}}^A, S_{\mathcal{PA}0}^A, \Sigma^A, \rightarrow_{\mathcal{PA}}^A, F_{\mathcal{PA}}^A)$  between  $\mathcal{T}_e^A/\sim$  and  $\mathcal{A}$  as described in Def. 5.2. Note that  $\mathcal{PA}^A$  is non-deterministic as  $\mathcal{T}_e^A/\sim$  is non-deterministic.

To finally solve Prob. 5.2, we formulate the fixed point problem:

$$J(s_{\mathcal{PA}}) = \min(J(s_{\mathcal{PA}}), \max_{s_{\mathcal{PA}} \rightarrow_{\mathcal{PA}}^A s'_{\mathcal{PA}}} J(s'_{\mathcal{PA}}) + 1), \quad (17)$$

initialized with  $J(s_{\mathcal{PA}}) = \infty$  for all  $s_{\mathcal{PA}} \in S_{\mathcal{PA}}^A \setminus F_{\mathcal{PA}}^A$  and  $J(s_{\mathcal{PA}}) = 0$  for all  $s_{\mathcal{PA}} \in F_{\mathcal{PA}}^A$ .

**Proposition 5.3:** Let  $S_{\mathcal{PA}}^{AS} = \{s_{\mathcal{PA}} \in S_{\mathcal{PA}}^A \mid J(s_{\mathcal{PA}}) < \infty\}$  and define  $\mathcal{X}^{AS} = \{eq(q) \mid (q, s) \in (S_{\mathcal{PA}0}^A \cap S_{\mathcal{PA}}^{AS})\}$ . Then  $\mathcal{X}^{AS}$  solves Prob. 5.2.

*Proof:* For each  $x \in \mathcal{X}^{AS}$ , there exists  $(q_0, s_0) \in S_{\mathcal{P}^A}^{AS}$  such that  $x \in \text{eq}(q_0)$  and  $s_0 \in S_{A0}$ . The fixed point algorithm guarantees that every trajectory of  $\mathcal{P}^A$  originating at  $(q_0, s_0)$  reaches  $F_{\mathcal{P}^A}^A$ . Then, construction of  $\mathcal{P}^A$  and bisimulation relation guarantee that all of the trajectories of (7) originating in  $\text{eq}(q_0)$  satisfy  $\phi$ .

If  $x_0 \notin \mathcal{X}^{AS}$ , we need to show that there exists a trajectory  $\mathbf{x} = x_0 \dots x_d$  of (7) that violates  $\phi$ . Let  $x_0 \in \text{eq}(q_0)$ . If  $x_0 \notin \mathcal{X}^{AS}$ , then for all  $s_0 \in S_{A0}$  there exists a trajectory  $\mathbf{p} = (q_0, s_0) \dots (q_d, s_d)$  of  $\mathcal{P}^A$  that can not reach  $F_{\mathcal{P}^A}^A$ , otherwise  $(q_0, s_0)$  would be included in  $S_{\mathcal{P}^A}^{AS}$ . Since  $\mathbf{p}$  can not reach  $F_{\mathcal{P}^A}^A$ ,  $\mathbf{q} = \gamma\tau(\mathbf{p})$  violates  $\phi$ . By bisimulation property, there exists a trajectory  $\mathbf{x} = x_0 \dots x_d$  of (7) that produces the same word as  $\mathbf{q}$ , and hence  $\mathbf{x}$  violates  $\phi$ . ■

**Example 5.3 (Example 5.1 continued):** For the example specification  $\phi$  as in (14), we obtained the solution to Prob. 5.2.  $\mathcal{X}^{AS}$  and sample trajectories are shown in Fig. 4. Note that this is a subset of the set of initial states found for the synthesis problem (see Fig.3).

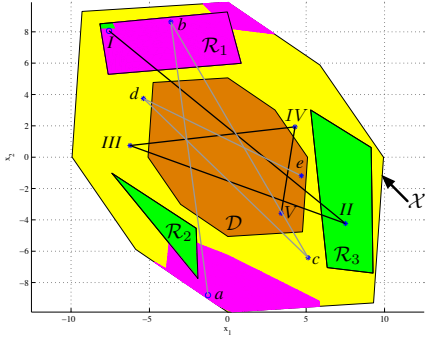


Fig. 4:  $\mathcal{X}^{AS}$  is shown in purple.  $\mathcal{X}$ ,  $\mathcal{D}$ ,  $\{\mathcal{R}_i\}_{i \in R}$  and two sample trajectories are indicated by labeling.

**Remark 5.1 (Implementation):** The methods described in this paper were implemented in MATLAB as a software package<sup>1</sup>, which uses the MPT toolbox [23] for polyhedral operations. Alg. 1 was completed in 2 hours for the example presented in the paper on an iMac with a Intel Core i5 processor at 2.8GHz with 8GB of memory. Once the bisimulation quotient is constructed, controller synthesis and verification were both completed in 2 minutes.

## VI. CONCLUSIONS

In this paper, we presented a method to abstract the behavior of a switched linear system within a positively invariant subset of  $\mathbb{R}^n$  to a finite transition system via the construction of a bisimulation quotient. We employed polyhedral Lyapunov functions to guide the partitioning of the state space and showed that the construction requires polytopic operations only. We showed how this method can be used to synthesize switching sequences and to verify the behavior of the system under arbitrary switching from specifications given as scLTL formulas over linear predicates in the state of the system.

<sup>1</sup>A preliminary version of the software is downloadable from [hyness.bu.edu/switchedpolybis.html](http://hyness.bu.edu/switchedpolybis.html).

## REFERENCES

- [1] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 61–70, 2007.
- [2] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multi-agent motion tasks based on LTL specifications," in *IEEE Conference on Decision and Control*, Paradise Islands, The Bahamas, 2004, pp. 153–158.
- [3] G. Batt, D. Ropers, H. De Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider, "Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in *escherichia coli*," *Bioinformatics*, vol. 21, no. suppl 1, pp. i19–i28, 2005.
- [4] R. Milner, *Communication and Concurrency*. Prentice-Hall, 1989.
- [5] M. C. Browne, E. M. Clarke, and O. Grumberg, "Characterizing finite kripke structures in propositional temporal logic," *Theoretical Computer Science*, vol. 59, no. 1-2, pp. 115–131, 1988.
- [6] J. M. Davoren and A. Nerode, "Logics for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 985–1010, 2000.
- [7] P. Tabuada and G. J. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1862–1877, 2006.
- [8] A. Chutinan and B. H. Krogh, "Verification of infinite-state dynamic systems using approximate quotient transition systems," *Automatic Control, IEEE Transactions on*, vol. 46, no. 9, pp. 1401–1410, 2001.
- [9] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical computer science*, vol. 126, no. 2, pp. 183–235, 1994.
- [10] B. Yordanov and C. Belta, "Formal analysis of discrete-time piecewise affine systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2834–2840, 2010.
- [11] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 116–126, 2010.
- [12] M. Lazar, "On infinity norms as Lyapunov functions: Alternative necessary and sufficient conditions," in *IEEE Conference on Decision and Control*, Atlanta, GA, 2010, pp. 5936–5942.
- [13] X. C. Ding, M. Lazar, and C. Belta, "Formal abstraction of linear systems via polyhedral Lyapunov functions," in *IFAC Conference on Analysis and Design of Hybrid Systems*, Eindhoven, The Netherlands, June 2012, to appear.
- [14] C. Sloth and R. Wisniewski, "Verification of continuous dynamical systems by timed automata," *Formal Methods in System Design*, vol. 39, pp. 47–82, 2011.
- [15] B. Grünbaum, *Convex polytopes*. Springer Verlag, 2003, vol. 221.
- [16] Z. P. Jiang and Y. Wang, "A converse Lyapunov theorem for discrete-time systems with disturbances," *Systems & control letters*, vol. 45, no. 1, pp. 49–58, 2002.
- [17] M. Lazar, "Model predictive control of hybrid systems: Stability and robustness," Ph.D. dissertation, Eindhoven University of Technology, 2006.
- [18] F. Blanchini, "Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 428–433, 1994.
- [19] J. Bochnak, M. Coste, and M. F. Roy, *Real algebraic geometry*. Springer Verlag, 1998, vol. 36.
- [20] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Formal Methods in System Design*, vol. 19, pp. 291–314, 2001.
- [21] E. M. Clarke, D. Peled, and O. Grumberg, *Model checking*. MIT Press, 1999.
- [22] T. Latvala, "Efficient model checking of safety properties," in *In Model Checking Software. 10th International SPIN Workshop*. Springer, 2003, pp. 74–88.
- [23] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: <http://control.ee.ethz.ch/~mpt/>